INSTITUTE FOR HOMELAND SECURITY

Feature Selection with Random Forest for Ransomware Detection

Qingzhong Liu

Feature Selection with Random Forest for Ransomware Detection

Qingzhong Liu
Department of Computer Science
Sam Houston State University
Email: liu@shsu.edu

Abstract

Ransomware continues to be a significant cybersecurity threat, requiring advanced detection techniques to mitigate its impact. In this study, we investigate the effectiveness of feature selection using the Random Forest machine learning algorithm for ransomware detection. Two datasets were analyzed: a large-scale Android ransomware dataset from Kaggle and the Ransomware Dataset 2024, containing various malware families with a strong focus on ransomware including notorious strains such as Cerber, REvil, and WannaCry. Feature selection was conducted using Random Forest's feature importance ranking, and multiple classification experiments were performed to evaluate model performance.

For **Dataset 1**, experimental results demonstrated that an optimal subset of features maintained high classification accuracy while preventing performance degradation due to redundant or irrelevant features. Feature Set 1 achieved the highest accuracy (99.73%), while Feature Set 5 provided the best balance between accuracy (99.09%) and model stability. The study highlights that adding excessive features beyond this threshold introduces redundancy and reduces performance.

For **Dataset 2**, both **binary classification** (benign vs. malicious), five**-category classification** (benign, ransomware, Trojan, spyware, and adware) and **27 family members** among the five categories were evaluated. The binary classification model achieved peak performance at Feature Set 3, with an accuracy of 99.45%, while the five-category classification attained a maximum accuracy of 95.91% at Feature Set 5, and the 27-family member classification peaked at Feature set 5, with the accuracy of 91.02%. The results confirm that feature selection plays a crucial role in improving detection performance and model efficiency.

Overall, this study demonstrates the potential of Random Forest-based feature selection in enhancing ransomware detection. Future research should explore deep learning-based methods for feature extraction and investigate real-time detection capabilities in dynamic network environments.

I. Introduction

Ransomware attacks have emerged as one of the most significant cybersecurity threats facing organizations and individuals in the digital age. These attacks encrypt victim data and demand payment for decryption, causing substantial financial losses and operational disruptions across sectors. The increasing sophistication of ransomware variants has driven extensive research into detection mechanisms that can identify and mitigate attacks before encryption completes. Below we examine the current state of ransomware detection research, briefly present below.

Behavioral Analysis Approaches

Behavioral analysis focuses on identifying ransomware by monitoring system activity patterns rather than relying on signature-based detection methods. This approach has gained prominence due to its potential for detecting previously unknown (zero-day) ransomware variants.

Continella et al. (2016) introduced ShieldFS, a system that builds models of normal file system activity and detects anomalies indicative of ransomware operations. ShieldFS maintains shadow copies of files being accessed, enabling recovery if malicious activity is detected.

Building on behavioral analysis, Mehnaz et al. (2018) developed RWGuard, which monitors file system operations at the kernel level. Their system analyzes sequences of API calls and I/O request patterns to distinguish between legitimate software and ransomware, achieving a 94% detection rate against various ransomware families.

Machine Learning and Deep Learning Approaches

Machine learning techniques have demonstrated considerable promise in ransomware detection by identifying patterns that might not be apparent through manual analysis.

Vinayakumar et al. (2019) made comprehensive evaluation of experiments of deep neural networks (DNNs) and other classical machine learning classifiers on various publicly available benchmark malware datasets. The optimal network parameters and network topologies for DNNs are chosen through the following hyperparameter selection methods with KDDCup 99 dataset. Through a rigorous experimental testing, it is confirmed that DNNs perform well in comparison with the classical machine learning classifiers.

Deep learning approaches have shown increasingly promising results. Chen et al. (2018) implemented a deep neural network model that analyzed byte-level file operations to detect ransomware activity across 2721 ransomware samples covering the majority of ransomware families.

Network Traffic Analysis

Network traffic analysis focuses on identifying ransomware by monitoring communication patterns, which can reveal command-and-control interactions and data exfiltration attempts before encryption begins.

Based on the observation of network communication of two crypto ransomware families, namely CryptoWall and Locky, Cabaj et al. (2018) analyzed the HTTP messages' sequences and their respective content sizes. The authors showed feasibility by designing and evaluating the proof-of-concept SDNbased detection system. Experimental results confirm that the proposed approach is feasible and efficient.

Crypto-Primitive Detection Approaches

Several researchers have focused on detecting cryptographic operations that are fundamental to ransomware functionality.

Kharraz et al. (2015) analyzed 1,359 samples that belong to 15 different ransomware families. The results show that, despite a continuous improvement in the encryption, deletion, and communication techniques in the main ransomware families, the number of families with sophisticated destructive capabilities remains quite small. In fact, our analysis reveals that in many samples, the malware simply locks the victim's computer desktop or attempts to encrypt or delete the victim's files using only superficial techniques. The experiments suggests that stopping advanced ransomware attacks is not as complex as it has been previously reported..

Scaife et al. (2016) developed CryptoDrop, an early-warning system designed to detect and prevent ransomware attacks. By monitoring file activity for suspicious behavior, CryptoDrop can interrupt processes that attempt to encrypt large volumes of user data. Leveraging specific ransomware indicators, the system achieves rapid detection with minimal false alarms. In the tests, CryptoDrop effectively stopped ransomware, resulting in a median data loss of only 10 files out of approximately 5,100. These results demonstrate that analyzing ransomware behavior enables the creation of highly effective detection systems, significantly reducing data loss for victims.

Hybrid and Multi-layered Approaches

Recent research has increasingly focused on hybrid approaches that combine multiple detection techniques to improve accuracy and reduce false positives.

Almashhadani et al. (2019) conducted a comprehensive behavioral analysis of crypto ransomware network activities, taking Locky, one of the most serious families, as a case study. A dedicated testbed was built, and a set of valuable and informative network features were extracted and classified into multiple types. A network-based intrusion detection system was implemented, employing two independent classifiers working in parallel on different levels: packet and flow levels. The experimental evaluation of the proposed detection system demonstrates that it offers high detection accuracy, low false positive rate, valid extracted features, and is highly effective in tracking ransomware network activities.

Unfortunately, all these studies rarely examine detection performance with different feature sets, especially different feature sets may lead to different detection results. Additionally, some datasets in previous study are small. To address these weaknesses and vulnerability, in this study,

we examine the feature selection with random forest machine learning for ransomware detection on two latest and significant datasets.

II. Data Sets in Our Study

1. The first dataset comes from Kaggle (Chakraborty 2023). It contains 203556 rows and 85 columns, and the entire data has 10 types of Android Ransomware and Benign traffic types. The type of Ransomware includes SVpeng, PornDroid, Koler, RansomBO, Charger, Simplocker, WannaLocker, Jisut, Lockerpin and Pletor, wherein:

   SVpeng Label contains 54161 Records
   PornDroid Label contains 46082 Records
   Koler Label contains 44555 Records
   Benign Label contains 43091 Records
   RansomBO Label contains 39859 Records
   Charger Label contains 39551 Records
   Simplocker Label contains 36340 Records
   WannaLocker Label contains 32701 Records
   Jisut Label contains 25672 Records
   Lockerpin Label contains 25307 Records
   Pletor Label contains 4715 Records
   The features include Flow ID, Source IP, Source Port Number, Destination IP, Destination Port Number, Protocol, Flow Duration, Total Fwd Packets, etc.
2. The dataset Ransomware Dataset 2024 (Amjad Hussain 2024) includes both malicious and benign samples, providing a balanced total of 21,752 samples, with 10,876 malicious and 10,876 benign files. The dataset is divided into five categories: one benign and four malicious categories **Trojan**, **Ransomware**, **Spyware**, and **Adware.** The dataset contains 27 distinct families: one benign and 26 distinct malware families, with a strong focus on ransomware, which includes:

   a. **Cerber**
   b. **DarkSide**
   c. **Dharma**
   d. **GandCrab**
   e. **LockBit**
   f. **Maze**
   g. **Phobos**
   h. **REvil**
   i. **Ragnar Locker**
   j. **Ryuk**
   k. **Shade**
   l. **WannaCry**

These 11 ransomware families represent some of the most notorious strains responsible for large-scale attacks in recent years. This dataset is valuable for advancing malware analysis, specifically in understanding ransomware behavior, and for building robust defenses against increasingly sophisticated attacks.

III. Our Approach: Random Forest and Feature Importance

Random Forest is an ensemble learning method that combines multiple decision trees to create a more robust and accurate predictive model (Breiman 2001).

Random Forest Algorithm

Random Forest builds on the concept of bagging (bootstrap aggregating) by creating many decision trees from random subsets of the training data and features. The final prediction is determined by aggregating the predictions of all trees.

The algorithm works as follows:

1. Create *n* bootstrap samples from the original dataset
2. For each bootstrap sample, grow a decision tree with the following modifications:
    - At each node, randomly select *m* features (where $m<p$, the total number of features)
    - Choose the best split from among these *m* features
    - Grow the tree to its maximum size without pruning
3. For classification, the final prediction is the majority vote of all trees
4. For regression, the final prediction is the average prediction of all trees

Mathematically, for a random forest with *B* trees and predictions for each tree:

Classification:

$$\hat{f}_{RF}(x) = majority\ vote\ \{f_b(x)\}_1^B \quad (1)$$

Regression:

$$\hat{f}_{RF} = \frac{1}{B}\Sigma_{b=1}^{B}\hat{f}_b(x) \quad (2)$$

Feature Importance Measures in Random Forest

Random Forest provides several methods to measure feature importance, here we introduce the method of mean decrease in impurity, also known as Gini importance or impurity-based importance, this measures the total decrease in node impurity (typically measured by Gini impurity for classification or variance for regression) weighted by the probability of reaching that node, averaged across all trees (Louppe et al. 2013; Strobl et al. 2007).

For feature X_j, the importance is:

$$\mathrm{Imp}(X_j) = \frac{1}{B}\sum_{b=1}^{B}\sum_{t\in T_b} p(t)\Delta i(s_t, X_j) \quad (3)$$

Where,

- T_b is the set of nodes in tree b
- $p(t)$ is the proportion of samples reaching node t
- $\Delta i(s_t, X_j)$ is the decrease in impurity when splitting on feature X_j at node t

IV. Experiments

Dataset 1

We applied a Random Forest learning classifier to Dataset 1 to evaluate malware detection performance. The optimal feature sets were selected based on feature importance, and twenty experiments were conducted for each detection scenario. Table 1 presents the nine feature sets selected through feature importance analysis, while Table 2 summarizes the mean and standard deviation values for accuracy, precision, recall, F1-score, and ROC-AUC across 20 experimental runs.

Table 1. Nine feature sets selected by feature importance on dataset 1.

Feature set	Features
1	[' Timestamp', ' Source IP', 'Flow ID', 'Unnamed: 0', ' Destination IP']
2	Feature set 1 +[' Source Port', ' Flow IAT Min', ' Flow IAT Max', ' Flow Duration', ' Flow Packets/s']
3	Feature set 2 + ['Fwd Packets/s', ' Flow IAT Mean', ' Fwd IAT Min', 'Init_Win_bytes_forward', ' Fwd IAT Max']
4	Feature set 3 + ['Fwd IAT Total', ' Fwd IAT Mean', ' Bwd Packets/s', ' Destination Port', ' Init_Win_bytes_backward']
5	Feature set 4 + ['Bytes/s', ' Flow IAT Std', ' Fwd IAT Std', ' Fwd Header Length', ' Fwd Header Length.1']
6	Feature set 5 + [' Average Packet Size', ' Avg Fwd Segment Size', ' Fwd Packet Length Mean', ' Packet Length Mean', ' Fwd Packet Length Max']
7	Feature set 6 + [' Packet Length Variance', ' Packet Length Std', ' Bwd IAT Min', ' Subflow Fwd Bytes', ' Avg Bwd Segment Size']
8	Feature set 7 + ['Total Length of Fwd Packets', ' Bwd Packet Length Mean', 'Bwd IAT Total', ' Bwd IAT Max', ' Bwd Header Length']
9	Feature set 8 + [' Total Length of Bwd Packets', ' Bwd IAT Mean', ' Subflow Bwd Bytes', ' Max Packet Length', ' Fwd Packet Length Std']

Table 2. The mean and standard deviation values of 20 experiments on Dataset 1 (%)

Feature Set	Accuracy (weighted)	Precision	Recall	F1-score	ROC-AUC
1	99.73/0.02	99.84/0.01	99.73/0.02	99.79/0.02	100.00/7.0e-06
2	99.70/0.02	99.89/0.01	99.70/0.02	99.80/0.02	100.00/1.4e-06
3	99.10/0.07	99.73/0.03	99.10/0.07	99.41/0.05	100.00/5.0e-06
4	99.13/0.05	99.74/0.03	99.13/0.05	99.44/0.03	100.00/4.8e-06
5	99.09/0.05	99.74/0.02	99.09/0.05	99.41/0.04	100.00/4.6e-06
6	98.41/0.05	99.58/0.03	98.41/0.05	98.98/0.04	99.99/8.0e-06
7	97.17/0.15	99.34/0.04	97.17/0.15	98.21/0.09	99.98/2.1e-05
8	97.42/0.11	99.38/0.03	97.42/0.11	98.37/0.07	99.98/1.3e-05
9	96.56/0.21	99.24/0.05	96.56/0.21	97.83/0.13	99.98/2.9-05

Based on the detection results in Table 2, we have the following observation:

The classification model demonstrated high accuracy across all feature sets, with values ranging from 99.73% (Feature Set 1) to 96.56% (Feature Set 9). The highest accuracy (99.73%) was observed in Feature Set 1, indicating that even a minimal set of features provided robust classification performance. Accuracy remained stable through Feature Set 5 (99.09%) but started declining from Feature Set 6 (98.41%) onward, reaching the lowest point at Feature Set 9 (96.56%). This downward trend suggests that adding more features beyond a certain threshold negatively impacted the model's effectiveness.

Precision remained consistently high across all feature sets, with values exceeding 99.2%, even as accuracy declined. The highest precision (99.89%) was recorded with Feature Set 2, indicating that additional flow-related features enhanced the model's ability to correctly classify malicious and benign samples. However, recall followed a trend similar to accuracy, starting at 99.73% in Feature Set 1 and gradually declining to 96.56% in Feature Set 9. The decrease in recall suggests that an increasing number of features introduced redundancy or noise, reducing the model's sensitivity.

The F1-score followed a similar pattern, peaking at 99.80% in Feature Set 2 before gradually declining to 97.83% in Feature Set 9. Since the F1-score represents the balance between precision and recall, the decline reflects the increasing difficulty in maintaining both as more features were introduced. Despite this, ROC-AUC values remained consistently near 100%, confirming the model's strong ability to distinguish between classes. However, the slight increase in standard deviation in later feature sets suggests growing variability in the classification results.

These findings indicate that while adding features initially improved performance, excessive inclusion led to diminishing returns and eventual degradation. Feature Set 1 demonstrated the highest accuracy (99.73%), showing that a minimal set of features was already effective. Feature Set 2 exhibited the highest precision (99.89%) and F1-score (99.80%), highlighting the impact of incorporating flow-related attributes. Feature Set 5 appeared to be the optimal balance point, maintaining high accuracy (99.09%), precision (99.74%), and recall (99.09%) before

performance began declining. The drop in accuracy and recall beyond Feature Set 5 suggests that additional features introduced redundancy rather than improving classification performance.

Based on these results, we recommend Feature Set 5 as the optimal feature set for Dataset 1. This set achieves high classification performance while avoiding the overfitting and instability observed in later feature sets.

Dataset 2

1. Binary Classification (Benign vs. Malicious Detection)

The following Table 3 lists the nine feature sets selected by feature importance and Table 4 shows each mean and standard deviation values over 20 experiments to detect benign and malicious classes.

Table 3. Nine feature sets selected by feature importance on dataset 2.

Feature set	Features
1	['processes_malicious', 'files_malicious', 'registry_total', 'registry_read', 'processes_monitored']
2	Feature set 1 + ['files_suspicious', 'network_dns', 'files_unknown', 'network_http', 'files_text']
3	Feature set 2 + ['registry_write', 'total_procsses', 'Subsystem', 'DllCharacteristics', 'network_connections']
4	Feature set 3 + ['AddressOfEntryPoint', 'apis', 'address_of_ne_header', 'processes_suspicious', 'rdata_SizeOfRawData']
5	Feature set 4 + ['rdata_VirtualAddress', 'rdata_VirtualSize', 'rdata_PointerToRawData', 'text_VirtualSize', 'EntryPoint']
6	Feature set 5 + ['SizeOfCode', 'SizeOfImage', 'PEType', 'MachineType', 'text_SizeOfRawData']
7	Feature set 6 + ['SizeOfInitializedData', 'BaseOfData', 'OperatingSystemVersion', 'dlls_calls', 'registry_delete']
8	Feature set 7 + ['Magic', 'md5', 'Checksum', 'FileAlignment', 'sha1']
9	Feature set 8 + ['bytes_on_last_page', 'SizeofHeapCommit', 'text_PointerToRawData', 'ImageVersion', 'SizeofStackReserve']

Table 4. The mean and standard deviation values of 20 experiments for benign and malicious detection (binary classification) on Dataset 2 (%)

Feature Set	Accuracy (weighted)	Precision	Recall	F1-score	ROC-AUC
1	99.14/0.12	99.16/0.12	99.14/0.12	99.15/0.12	99.73/0.08
2	99.25/0.11	99.26/0.11	99.25/0.11	99.26/0.11	99.87/0.05
3	99.45/0.10	99.47/0.09	99.45/0.09	99.46/0.09	99.94/0.02
4	99.44/0.09	99.45/0.09	99.44/0.09	99.44/0.09	99.96/0.02
5	99.41/0.09	99.43/0.09	99.41/0.10	99.42/0.10	99.96/0.02

6	99.40/0.10	99.42/0.11	99.40/0.11	99.41/0.11	99.96/0.03
7	99.43/0.10	99.45/0.11	99.43/0.10	98.44/0.11	99.96/0.02
8	99.41/0.11	99.43/0.11	99.41/0.11	99.42/0.11	99.96/0.02
9	99.40/0.11	99.43/0.11	96.40/0.11	99.41/0.11	99.96/0.02

The performance of the binary classification model was evaluated using nine different feature sets. The baseline model, using Feature Set 1, achieved a high accuracy of 99.14%, with precision, recall, and F1-score at similar levels. As additional features were introduced, there was a notable improvement in performance, particularly with Feature Set 3, where accuracy peaked at 99.45%. The inclusion of network-related features (network_dns, network_http, network_connections) and additional registry-related features significantly enhanced classification performance.

Beyond Feature Set 3, the model's accuracy, precision, recall, and F1-score showed minimal fluctuations, indicating that additional features did not contribute substantial improvements. The ROC-AUC metric consistently increased from 99.73% in Feature Set 1 to 99.96% by Feature Set 4, signifying improved distinction between benign and malicious samples. Importantly, the standard deviation remained low (~0.1 across all metrics), ensuring stable performance across multiple experiments.

The results suggest that while the initial feature set was already highly effective, Feature Set 3 provides the optimal balance of features for binary classification. Adding more features beyond this point does not yield meaningful improvements and may introduce unnecessary complexity.

2. Five-Category Classification

The following Table 5 lists the nine feature sets selected by feature importance and Table 6 shows each mean and standard deviation values over 20 experiments to detect the five category classes.

Table 5. Nine feature sets selected by feature importance on dataset 2 for category detection.

Feature set	Features
1	['processes_malicious', 'files_suspicious', 'files_unknown', 'files_malicious', 'files_text']
2	Feature set 1 + ['processes_monitored', 'registry_total', 'network_dns', 'registry_read', 'total_procsses']
3	Feature set 2 + ['network_http', 'DllCharacteristics', 'registry_write', 'AddressOfEntryPoint', 'OperatingSystemVersion']
4	Feature set 3 + ['network_connections', 'apis', 'SizeOfImage', 'rdata_PointerToRawData', 'rdata_VirtualAddress']
5	Feature set 4 + ['rdata_VirtualSize', 'text_VirtualSize', 'text_SizeOfRawData', 'dlls_calls', 'rdata_SizeOfRawData']
6	Feature set 5 + ['SizeOfCode', 'SizeOfInitializedData', 'address_of_ne_header', 'BaseOfData', 'Checksum']

7	Feature set 6 + ['Subsystem', 'EntryPoint', 'ImageVersion', 'text_PointerToRawData', 'SizeOfHeaders']
8	Feature set 7 + ['processes_suspicious', 'BaseOfCode', 'SectionAlignment', 'sha1', 'md5']
9	Feature set 8 + ['FileAlignment', 'SizeOfUninitializedData', 'text_VirtualAddress', 'SizeofStackReserve', 'bytes_on_last_page']

Table 6. The mean and standard deviation values of 20 experiments for five category classification on Dataset 2 (%)

Feature Set	Accuracy (weighted)	Precision	Recall	F1-score	ROC-AUC
1	87.45/0.47	92.84/0.54	87.45/0.47	89.84/0.29	97.84/0.12
2	93.93/0.36	97.22/0.26	93.93/0.36	95.50/0.29	99.45/0.07
3	95.47/0.10	98.62/0.16	95.47/0.17	97.00/0.13	99.80/0.03
4	95.90/0.17	98.93/0.11	95.90/0.17	97.36/0.10	99.85/0.03
5	95.91/0.18	98.93/0.16	95.91/0.18	97.36/0.15	99.85/0.02
6	95.72/0.19	98.99/0.11	95.72/0.19	97.30/0.13	99.86/0.02
7	95.70/0.20	99.06/0.15	95.71/0.20	97.30/0.16	99.87/0.02
8	95.66/0.19	99.04/0.13	95.66/0.19	97.29/0.15	99.86/0.02
9	95.63/0.18	99.04/0.15	95.64/0.18	97.27/0.15	99.86/0.02

For the five-category classification task, the baseline Feature Set 1 provided an accuracy of 87.45%, indicating that the initial set of features was insufficient for distinguishing multiple malware categories. A significant improvement was observed when transitioning to Feature Set 2, where accuracy increased to 93.93%. This suggests that adding process monitoring, registry features, and network-based features substantially enhances classification performance.

The model's accuracy continued to improve, reaching its peak at Feature Set 5 with 95.91% accuracy. After this point, additional features led to only marginal fluctuations, with no significant improvement in classification performance. Precision remained consistently high throughout the feature expansion process, exceeding 98.9% from Feature Set 3 onward, indicating that the model was highly confident in its classifications. The recall metric, while slightly lower, maintained strong performance, ensuring balanced sensitivity across categories.

ROC-AUC scores improved from 97.84% (Feature Set 1) to 99.86% (Feature Set 5), demonstrating enhanced separability between categories. However, after Feature Set 5, the additional features primarily consisted of cryptographic hashes and PE metadata, which did not provide further meaningful contributions.

Based on these findings, Feature Set 5 represents the optimal feature set for five-category classification, as it achieves the highest accuracy while maintaining a strong balance between precision and recall.

3. 27-Family Benign, Ransomware and Other Malware Classification

The following Table 7 lists the nine feature sets selected by feature importance and Table 8 shows each mean value and standard deviation values over 20 experiments to detect the 27 distinct family classes.

Table 7. Nine feature sets selected by feature importance on dataset 2 for category detection.

Feature set	Features
1	['processes_malicious', 'files_malicious', 'files_suspicious', 'registry_total', 'processes_monitored']
2	Feature set 1 + ['processes_monitored', 'files_text', 'registry_read', 'SizeOfImage', 'DllCharacteristics', 'network_dns']
3	Feature set 2 + ['network_http', 'files_unknown', 'total_procsses', 'rdata_VirtualSize', 'SizeOfInitializedData']
4	Feature set 3 + ['AddressOfEntryPoint', 'apis', 'address_of_ne_header', 'text_VirtualSize', 'rdata_SizeOfRawData']
5	Feature set 4 + ['rdata_PointerToRawData', 'rdata_VirtualAddress', 'network_connections', 'OperatingSystemVersion', 'dlls_calls']
6	Feature set 5 + ['EntryPoint', 'SizeOfCode', 'registry_write', 'text_SizeOfRawData', 'BaseOfData']
7	Feature set 6 + ['Checksum', 'sha1', 'text_PointerToRawData', 'md5', 'ImageVersion']
8	Feature set 7 + ['SizeOfHeaders', 'processes_suspicious', 'FileAlignment', 'ImageBase', 'SectionAlignment']
9	Feature set 8 + ['text_VirtualAddress', 'SizeOfUninitializedData', 'BaseOfCode', 'Subsystem', 'SizeofStackReserve']

Table 8. The mean and standard deviation values of 20 experiments for the 27 family member classification on Dataset 2 (%)

Feature Set	Accuracy (weighted)	Precision	Recall	F1-score	ROC-AUC
1	86.58/0.44	93.34/0.30	86.58/0.44	89.56/0.37	97.79/0.21
2	90.52/0.47	98.34/0.35	90.52/0.47	93.93/0.37	99.47/0.14
3	90.60/0.37	99.37/0.11	90.60/0.37	94.35/0.25	99.73/0.06
4	90.88/0.35	99.49/0.09	90.88/0.35	94.57/0.23	99.80/0.04
5	91.02/0.41	99.54/0.09	91.02/0.41	94.66/0.28	99.81/0.05
6	90.88/0.40	99.60/0.08	90.88/0.40	94.59/0.27	99.83/0.04
7	90.55/0.40	99.64/0.07	90.55/0.40	94.37/0.28	99.83/0.04
8	90.66/0.30	99.64/0.09	90.66/0.30	94.44/0.22	99.83/0.04
9	90.70/0.33	99.60/0.09	90.70/0.33	94.45/0.23	99.83/0.03

The 27-family malware classification task presented a more challenging scenario, with the baseline Feature Set 1 achieving an accuracy of 86.58%. This relatively lower accuracy indicates that the initial set of features was insufficient for distinguishing between malware families. A substantial improvement was observed with Feature Set 2, where accuracy increased to 90.52%,

highlighting the importance of registry, network, and PE metadata features in fine-grained malware classification.

Performance continued to improve, peaking at Feature Set 5 with 91.02% accuracy. Precision consistently increased across feature sets, reaching 99.64% by Feature Set 7, which suggests that the model was effective in minimizing false positives. However, recall remained slightly lower than precision throughout the experiments, peaking at 91.02% (Feature Set 5), indicating that some malware families were still misclassified.

The ROC-AUC metric improved from 97.79% in Feature Set 1 to 99.83% by Feature Set 5, demonstrating the model's increasing ability to distinguish between malware families. However, beyond Feature Set 5, additional features such as cryptographic hashes (md5, sha1) and other PE metadata did not provide significant improvements in classification performance. This suggests that these features might introduce redundancy rather than contributing new distinguishing information.

Given these findings, Feature Set 5 is the optimal feature set for 27-family classification, balancing accuracy and recall while ensuring robust family-level classification.

Across all three classification tasks, the results highlight the critical role of network, registry, and process-related features in malware detection. These features consistently contributed to significant performance improvements, particularly up to Feature Set 3 in binary classification and Feature Set 5 in multi-class classification tasks.

The experimental results also suggest that additional cryptographic hashes and PE metadata features beyond a certain point do not yield substantial accuracy gains. While they may provide some additional information, their contribution appears to be marginal compared to network and registry-based features.

Another key observation is the trade-off between precision and recall, particularly in the multi-class and family classification tasks. While precision consistently remained high, recall was comparatively lower, indicating that the model was confident in its classifications but occasionally misclassified rarer malware families. Future work could explore data augmentation or advanced ensemble techniques to improve recall for rare classes.

In summary, the optimal feature sets for each classification task are:

Binary classification: Feature Set 3 (Accuracy: 99.45%)
Five-category classification: Feature Set 5 (Accuracy: 95.91%)
27-family classification: Feature Set 5 (Accuracy: 91.02%)

These feature sets strike the best balance between classification performance and computational efficiency, making them recommended choices for future malware detection systems.

V. Conclusions

This study examined the effectiveness of feature selection using the Random Forest machine learning algorithm for ransomware detection. Two datasets were analyzed: Dataset 1, which contained Android ransomware and benign traffic samples, and Dataset 2, which consisted of benign and malicious samples categorized into multiple malware families. The experimental results demonstrated that feature selection plays a critical role in optimizing classification performance.

For Dataset 1, the highest accuracy (99.73%) was achieved using a minimal feature set, indicating that a small but well-selected set of features was sufficient for effective detection. However, beyond Feature Set 5, adding more features led to diminishing returns, reducing accuracy and recall due to redundancy and noise. Similarly, in Dataset 2, the best binary classification performance was achieved with Feature Set 3 (99.45% accuracy), while the five-category classification peaked with Feature Set 5 (95.91% accuracy). These findings highlight the importance of balancing feature selection to optimize performance while avoiding unnecessary complexity.

Overall, the study confirms that Random Forest, combined with effective feature selection, is a robust approach for ransomware detection. The high precision, recall, and ROC-AUC scores across both datasets demonstrate the model's capability to distinguish between benign and malicious samples with high reliability.

Further Study

Future research should explore the following areas to further improve ransomware detection:

1. **Real-Time Detection and Deployment** – Implementing and evaluating the model in real-world network environments to assess its performance under live traffic conditions.
2. **Feature Reduction Techniques** – Investigating dimensionality reduction methods, such as Principal Component Analysis (PCA) and Autoencoders, to optimize feature selection further.
3. **Comparison with Other Machine Learning Models** – Benchmarking Random Forest against other advanced models, such as deep learning and hybrid approaches, to determine the most effective detection framework.
4. **Generalization Across Ransomware Variants** – Testing the model on new and emerging ransomware families to evaluate its adaptability and robustness against evolving threats.
5. **Explainability and Interpretability** – Developing methods to interpret model decisions to enhance transparency and trust in automated malware detection systems.

By addressing these aspects, future studies can enhance the scalability, adaptability, and effectiveness of machine learning-based ransomware detection systems.

Acknowledgements

We are grateful to the SHSU Institute of Homeland Security for supporting this study.

References

Breiman, L. (2001). Random Forests. Machine Learning, 45(1), 5-32.

Louppe, G., Wehenkel, L., Sutera, A., & Geurts, P. (2013). Understanding variable importances in forests of randomized trees. Advances in Neural Information Processing Systems, 26.

Strobl, C., Boulesteix, A. L., Zeileis, A., & Hothorn, T. (2007). Bias in random forest variable importance measures: Illustrations, sources and a solution. BMC Bioinformatics, 8(1), 25.

Chakraborty, S. (2023). Android Ransomware Detection [Data set]. Kaggle. https://doi.org/10.34740/KAGGLE/DSV/4987535

Amjad Hussain, A. H. (2024). Ransomware Dataset 2024 (1.0) [Data set]. Zenodo. https://doi.org/10.5281/zenodo.13890887

Almashhadani, A. O., Kaiiali, M., Sezer, S., & O'Kane, P. (2019). A multi-classifier network-based crypto ransomware detection system: A case study of locky ransomware. IEEE access, 7, 47053-47067.

Cabaj, K., Gregorczyk, M., & Mazurczyk, W. (2018). Software-defined networking-based crypto ransomware detection using HTTP traffic characteristics. Computers & Electrical Engineering, 66, 353-368.

Chen, J., Wang, C., Zhao, Z., Chen, K., Du, R., & Ahn, G. J. (2018). Uncovering the face of android ransomware: Characterization and real-time detection. IEEE Transactions on Information Forensics and Security, 13(5), 1286-1300.

Continella, A., Guagnelli, A., Zingaro, G., De Pasquale, G., Barenghi, A., Zanero, S., & Maggi, F. (2016). ShieldFS: A self-healing, ransomware-aware filesystem. Proceedings of the 32nd Annual Conference on Computer Security Applications, 336-347.

Kharraz, A., Robertson, W., Balzarotti, D., Bilge, L., & Kirda, E. (2015). Cutting the gordian knot: A look under the hood of ransomware attacks. In Detection of Intrusions and Malware, and Vulnerability Assessment: 12th International Conference, DIMVA 2015, Milan, Italy, July 9-10, 2015, Proceedings 12 (pp. 3-24). Springer International Publishing.

Mehnaz, S., Mudgerikar, A., & Bertino, E. (2018, September). Rwguard: A real-time detection system against cryptographic ransomware. In International Symposium on Research in Attacks, Intrusions, and Defenses (pp. 114-136). Cham: Springer International Publishing.

Scaife, N., Carter, H., Traynor, P., & Butler, K. R. (2016, June). Cryptolock (and drop it): stopping ransomware attacks on user data. In 2016 IEEE 36th international conference on distributed computing systems (ICDCS) (pp. 303-312). IEEE.

Vinayakumar, R., Alazab, M., Soman, K. P., Poornachandran, P., Al-Nemrat, A., & Venkatraman, S. (2019). Deep learning approach for intelligent intrusion detection system. IEEE access, 7, 41525-41550.

INSTITUTE FOR HOMELAND SECURITY

The Institute for Homeland Security at Sam Houston State University is focused on building strategic partnerships between public and private organizations through education and applied research ventures in the critical infrastructure sectors of Transportation, Energy, Chemical, Water / Wastewater, Healthcare, and Public Health.

The Institute is a center for strategic thought with the goal of contributing to the security, resilience, and business continuity of these sectors from a Texas Homeland Security perspective. This is accomplished by facilitating collaboration activities, offering education programs, and conducting research to enhance the skills of practitioners specific to natural and human caused Homeland Security events.

Institute for Homeland Security
Sam Houston State University

Liu, Q. (2025). Feature Selection with Random Forest for Ransomware Detection (Institute for Homeland Security Report No. 2025-1008). Institute for Homeland Security.
https://doi.org/10.17605/OSF.IO/PN8TQ